α and β Stability for Additively Regularized LS-SVMs via Convex Optimization

K. Pelckmans, J.A.K. Suykens, B. De Moor
K.U. Leuven - ESAT - SCD/SISTA
Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee) - Belgium
Phone: +32-16-32 85 40, Fax: +32-16-32 19 70
E-mail: {kristiaan.pelckmans, johan.suykens}@esat.kuleuven.ac.be
Web: http://www.esat.kuleuven.ac.be/sista/lssvmlab/

Abstract

This paper considers the design of an algorithm that maximizes explicitly its own stability. The stability criterion - as often used for the construction of bounds on the generalization error of a learning algorithm - is proposed to compensate for overfitting. The primal-dual formulation characterizing Least Squares Support Vector Machines (LS-SVMs) and the additive regularization framework [13] are employed to derive a computational and practical approach combined with convex optimization. The method is elaborated for non-linear regression as well as classification. The proposed stable kernel machines also lead to a new notion of L_α and L_β curves instead of the traditional L-curves defined on training data.
keywords: Kernel machines, Support Vector Machines, Stability.

1 Introduction

Regularization has a rich history which dates back to the theory of inverse ill-posed and ill-conditioned problems [21]. It has influenced research in modeling such as regularized cost functions as occurring e.g. in splines [23], multilayer-perceptrons [1], regularization networks [14], support vector machines (SVM) and related methods (see e.g. [10]). SVM [22] is a powerful methodology for solving problems in linear and nonlinear classification, function estimation and density estimation which has also led to many other recent developments in kernel based learning methods in general [6, 17]. SVMs and related methods have been introduced within the context of statistical learning theory and structural risk minimization. In the methods one solves convex optimization problems, typically quadratic programs. Least Squares Support Vector Machines (LS-SVMs)[1] [20, 19] are reformulations to standard SVMs which lead to solving linear KKT systems for classification tasks as well as regression. In [19] LS-SVMs have been proposed as a class of kernel machines with primal-dual formulations in relation to kernel FDA, RR, PLS, CCA, PCA, recurrent networks and control. The dual problems for the static regression without bias term are closely related to Gaussian processes [11], regularization networks [14] and Kriging [5], while LS-SVMs rather take an optimization approach with primal-dual formulations and application of the kernel trick.

Sensitivity analysis aims at determining how much the variation of the input (data) can influence the output of a system. This notion is used in many different domains (numerical, robust statistics, control theory) under different denominators (e.g. sensitivity, perturbation, influence or conditioning). The more specific definition of stability of a learning algorithm defined in e.g. [7, 2, 15] is used in this paper. Originally, it was proposed for the estimation of the accuracy of learning algorithms itself by revealing the connection between stability and generalization error [7]. In particular, one can derive [2] a bound on the generalization error or risk functional based on an observed quantitative measure of stability. Although many subtle differences exist between different definitions (one distinguishes amongst others between (pointwise) hypothesis, error or uniform stability), this paper only works with the two concepts of uniform α and

[1] http://www.esat.kuleuven.ac.be/sista/lssvmlab

β stability as they are most clearly put within an optimization point of view. Uniform stability was used to derive exponential bounds for different algorithms, including techniques for unsupervised learning (k-nearest neighbor), classification (soft margin SVMs) and regression (Regularized least squares regression or LS-SVMs).

While in previous papers about stability, the object of interest was the learning algorithm itself [2], this paper considers the design of an algorithm optimizing its own stability. The primal-dual formulations characterizing Least Squares Support Vector Machines (LS-SVMs) [19] and the additive regularization framework [13] are employed to derive a practical approach. The result is a kernel method where structure is imposed on the model class by using a measure of stability as a complexity term. This paper differentiates between α, β and 2-norm stability all based on leave-one-out cross-validation, which results in convex optimization problems or even a set of linear equations. The method is elaborated for non-linear regression as well as classification. The selection of the trade-off between model fit and model complexity (commonly referred to as the regularization constant) is facilitated by the introduction of the L_α- and the L_β-curve as alternatives for the traditional L-curve in numerical linear algebra [12, 8, 4].

This paper is organized as follows: the following section reviews primal dual derivations of LS-SVMs for regression and classification together with the idea of fusion of training and validation via the alternative regularization scheme called additive regularization. This section is mainly based on results from the paper [13]. Section 3 considers the extension to fusion of additively regularized LS-SVMs towards a leave-one-out cross-validation setting and introduces an extension towards stability measures for the design of α-stable and β-stable machines. Section 4 gives a toy example which illustrates the practical relevance of results in this paper.

2 Primal Dual Formulations and Fusion

This section reviews the estimation of LS-SVMs and the automatic tuning of the regularization trade-off with respect to a validation measure. As proposed in [13], fusion of training and validation levels can be investigated from an optimization point of view, while conceptually, they are considered at different levels. Although this paper only reports result in the case of regression, extensions to classification are straightforward.

The following notation is used throughout this paper. Giving a training set $\mathcal{D}_N = \{x_k, y_k\}_{k=1}^N \subset \mathbb{R}^D \times \mathbb{R}$ of size N drawn i.i.d. from an unknown distribution F_{XY} according to $y_k = f(x_k) + e_k$ where $f : \mathbb{R}^D \to \mathbb{R}$ is an unknown real-valued smooth function, $E[y_k|X = x_k] = f(x_k)$ and $e_1, \ldots, e_N$ are uncorrelated random errors with $E\left[e_k|X = x_k\right] = 0$, $E\left[(e_k)^2|X = x_k\right] = \sigma_e^2 < \infty$. The n data points of the validation set are denoted as $\mathcal{D}_n^{(v)} = \{x_j^{(v)}, y_j^{(v)}\}_{j=1}^n$. The following vector notations are used throughout the text: $X = (x_1, \ldots, x_N) \in \mathbb{R}^{D\times N}$, $Y = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$, $X^{(v)} = \left(x_1^{(v)}, \ldots, x_n^{(v)}\right) \in \mathbb{R}^{D\times n}$ and $Y^{(v)} = \left(y_1^{(v)}, \ldots, y_n^{(v)}\right)^T \in \mathbb{R}^n$.

2.1 Standard LS-SVM Regressors and Classifiers

Consider first the regression case. The model of a LS-SVM regressor is given as $f(x) = w^T\varphi(x) + b$ in the primal space where $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_f}$ denotes the potentially infinite ($d_f = \infty$) dimensional feature map. The regularized least squares cost function is given as [19]

$$\min_{w,b,e_k} \mathcal{J}_\gamma(w, e) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{k=1}^N e_k^2 \text{ s.t. } w^T\varphi(x_k) + b + e_k = y_k, \quad k = 1, \ldots, N. \tag{1}$$

Note that the regularization constant γ appears here as in classical Tikhonov regularization [21]. The solution corresponds with a form of ridge regression [16], regularization networks [14], Gaussian processes [11] and Kriging [5], but usually considers a bias term b and formulates the problem in a primal-dual

optimization context. The Lagrangian of the constrained optimization problem becomes

$$\mathcal{L}_\gamma(w, b, e_k; \alpha_k) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N} \alpha_k (w^T \varphi(x_k) + b + e_k - y_k). \tag{2}$$

By taking the conditions for optimality $\partial \mathcal{L}_\gamma / \partial \alpha_k = 0$, $\partial \mathcal{L}_\gamma / \partial b = 0$, $\partial \mathcal{L}_\gamma / \partial e_k = 0$ and $\partial \mathcal{L}_\gamma / \partial w = 0$ and application of the kernel trick $K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l)$ for all $k, l = 1, \ldots, N$ with a positive definite (Mercer) kernel K, one gets the following conditions for optimality

$$\left\{ \begin{array}{rll} y_k & = w^T \varphi(x_k) + b + e_k, & k = 1, \ldots, N \quad (a) \\ 0 & = \sum_{k=1}^{N} \alpha_k & \quad (b) \\ e_k \gamma & = \alpha_k, & k = 1, \ldots, N \quad (c) \\ w & = \sum_{k=1}^{N} \alpha_k \varphi(x_k). & \quad (d) \end{array} \right. \tag{3}$$

The dual problem is summarized as follows after elimination of the variables e_k and w

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + I_N / \gamma \end{array} \right] \left[\begin{array}{c} b \\ \hline \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \hline Y \end{array} \right], \tag{4}$$

where $\Omega \in \mathbb{R}^{N \times N}$ with $\Omega_{kl} = K(x_k, x_l)$ and $Y = (y_1, \ldots, y_N)^T$. The estimated function $\hat{f}$ can be evaluated at a new point x^* by

$$\hat{f}(x_*) = \sum_{k=1}^{N} \alpha_k K(x_k, x_*) + b, \tag{5}$$

for α and b solving (4).

A derivation of LS-SVMs was given originally for the binary classification task [20]. In this case, the output labels take values -1 or 1. The LS-SVM classifier $\mathrm{sign}(w^T \varphi(x) + b)$ is optimized with respect to

$$\min_{w,b,e_k} \mathcal{J}_\gamma(w, e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{k=1}^{N} e_k^2 \text{ s.t. } y_k (w^T \varphi(x_k) + b) = 1 - e_k, \quad \forall k = 1, \ldots, N. \tag{6}$$

Using a primal dual optimization interpretation, the unknowns α, b of the estimated classifier $\mathrm{sign}(\sum_{k=1}^{N} \alpha_k y_k K(x_k, x) + b)$ are found by solving the dual set of linear equations

$$\left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega^y + I_N / \gamma \end{array} \right] \left[\begin{array}{c} b \\ \hline \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \hline 1_N \end{array} \right] \tag{7}$$

where $\Omega^y \in \mathbb{R}^{N \times N}$ with $\Omega^y_{ij} = y_i y_j K(x_i, x_j)$. The remainder focuses on the regression case, although it is applicable just as well to the classification problem.

2.2 Fusion of Standard LS-SVMs and Validation for Regularization Constant Tuning

The fusion argument as introduced in [13] is briefly revised in relation to regularization parameter tuning. The LS-SVM regressor on the training data for a fixed value γ is given as (1)

$$\textbf{Level 1}: \ (\hat{w}, \hat{b}, \hat{e}) = \arg\min_{w,b,e} \mathcal{J}_\gamma(w, e) \quad \text{s.t. (1) holds}, \tag{8}$$

which results into conditions for optimality summarized as a linear set of equations (4) after substitution of w by Lagrange multipliers α. Tuning the regularization parameter by using a validation criterion gives the following estimator

$$\textbf{Level 2}: \ \hat{\gamma} = \arg\min_{\gamma} \sum_{j=1}^{n} \left(f(x_j^{(v)}; \hat{w}, \hat{b}, \hat{e}) - y_j^{(v)} \right)^2 \quad \text{with} \quad (\hat{w}, \hat{b}, \hat{e}) = \arg\min_{w,b,e} \mathcal{J}_\gamma \tag{9}$$

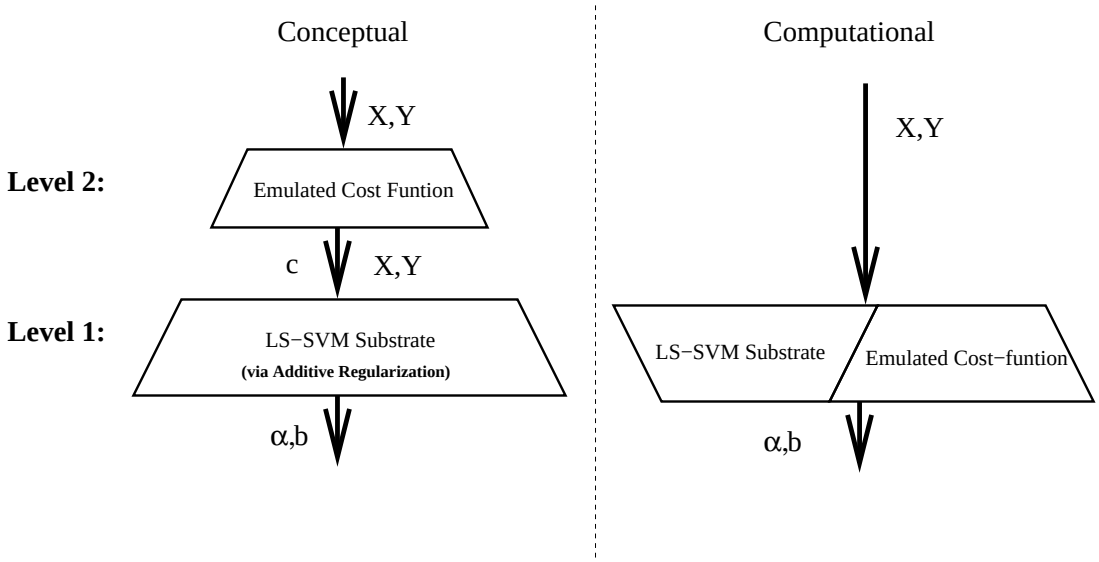


Figure 1: *Graphical representation of the additive regularization framework used for emulating other loss functions and regularization schemes. Conceptually, one differentiates between the newly specified cost function (at level 2) and the LS-SVM substrate (at level 1), while computationally both are obtained simultaneously.*

satisfying again (1). Using the conditions for optimality (4) and eliminating w and e

$$\textbf{Fusion}: \ (\hat{\gamma}, \hat{\alpha}, \hat{b}) = \underset{\gamma,\alpha,b}{\arg\min} \sum_{j=1}^{n} \left(f(x_j^{(v)}; \alpha, b) - y_j^{(v)} \right)^2 \quad \text{s.t.} \quad (4) \text{ holds}, \tag{10}$$

which is referred to as fusion. The resulting optimization problem (10) is usually non-convex. The optimal solutions w or dual α's corresponding with a $\gamma > 0$ belong to a non-convex set in search-space.

2.3 Fusion of Additively Regularized LS-SVMs and Validation

To overcome this problem, a re-parameterization of the trade-off was proposed leading to the so-called additive regularization scheme introduced in [13]. Let the trade-off be parameterized using a vector $c = (c_1, \ldots, c_N)^T$ of hyper-parameters such that

$$\textbf{Level 1}: \ (\hat{w}, \hat{b}, \hat{e}) = \underset{w,b,e}{\arg\min} \mathcal{J}_c(w, e) = \frac{1}{2} w^T w + \frac{1}{2} \sum_{k=1}^{N} (e_k - c_k)^2$$
$$\text{s.t.} \quad w^T \varphi(x_k) + b + e_k = y_k, \quad k = 1, \ldots, N \tag{11}$$

where the conditions of optimality can be summarized as (see Subsection 2.1 and [13]) after elimination of w and e

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline c \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y \end{array}\right]. \tag{12}$$

At the cost of over-parameterizing the trade-off, the regularization constants c appear linearly in the set of equations. To circumvent overfitting on c, different ways to restrict explicitly or implicitly the (effective) degrees of freedom of the regularization scheme $c \in \mathcal{A} \subset \mathbb{R}^N$ were proposed in [13].

For example, cost function (1) is a special case of the additive regularization cost function (11) as the conditions for optimality of both (4) and (12) are equal when adding the constraint $\alpha + c = \gamma^{-1}\alpha$

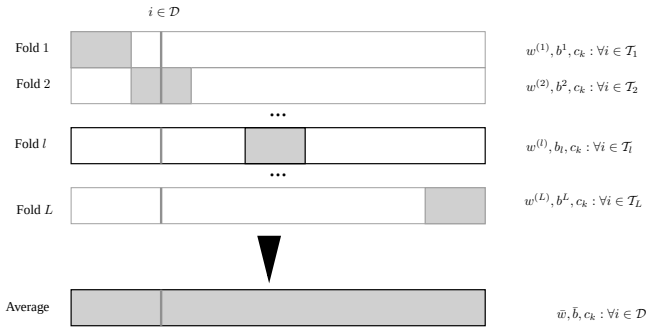


Figure 2: *Schematical illustration of the L-fold cross-validation procedure.*

to the latter. Let the bias term b be omitted in the following remark. In general, every kernel machine $f(x) = w^T \varphi(x)$ which was optimized in some (regularized) sense on the training dataset $\mathcal{D}$

$$\mathcal{J}_{\text{km}}(w, e_k) = \frac{1}{2} w^T w + \gamma \sum_{k=1}^{N} \ell(e_k) \tag{13}$$

for any convex loss function $\ell : \mathbb{R} \to \mathbb{R}$, can be written in the form $\hat{f}(x) = \sum_{k=1}^{N} \hat{\alpha}_k K(x_k, x)$ with $\hat{\alpha} \in \mathbb{R}^N$ (application of representer theorem [22]). As such, every kernel machine (for regression) can be written as a special case of the additively regularized LS-SVM (11) in combination with an appropriate subset $\mathcal{A} \subset \mathbb{R}^N$ for c. The latter can be recovered from the equality $(\Omega + I_N)\hat{\alpha} + c = Y$ (obtained as conditions for optimality (12) without a bias term) for all solutions $\hat{\alpha}$ optimizing (13) for positive values of ξ. Thus, the determination of an appropriate subset $\mathcal{A}$ is to be considered as model selection in a broad sense.

Let us now fix an appropriate subset $\mathcal{A}$. Tuning the regularization parameter by using a validation criterion results in the following optimization problem

$$\textbf{Level 2}: \ \hat{\gamma} = \underset{c \in \mathcal{A}}{\arg\min} \sum_{j=1}^{n} \left(f(x_j^{(v)}; \hat{\alpha}, \hat{b}) - y_j^{(v)} \right)^2 \quad \text{with} \quad (\hat{\alpha}, \hat{b}) = \underset{\alpha, b}{\arg\min}\, \mathcal{J}_c. \tag{14}$$

By substitution of the constraints in (11) by their corresponding conditions for optimality (12), one can write

$$\textbf{Fusion}: \ (\hat{c}, \hat{\alpha}, \hat{b}) = \underset{c \in \mathcal{A}, \alpha, b}{\arg\min} \sum_{j=1}^{n} \left(f(x_j^{(v)}; \alpha, b) - y_j^{(v)} \right)^2 \quad \text{s.t.} \quad (12) \text{ holds}. \tag{15}$$

This can be solved efficiently as a convex optimization problem if $\mathcal{A}$ is a convex set, resulting immediately in the globally optimal regularization trade-off and model parameters [3].

3 Fusion with Leave-one-out Based Criteria

This section considers a special case of the previous argument where a regularization scheme (or an appropriate set $\mathcal{A}$) is designed to be used in a leave-one-out framework [13]. The cross-validation procedure [18] is described in Figure 2. Let a superscript $^{(l)}$ for $l = 1, \ldots, L$ denote the unknowns associated with the lth fold. Let $\mathcal{T}^{(l)}$ denote the set of data used for training in the lth fold and let $\mathcal{V}^{(l)}$ denote the validation data of the lth fold. The new result of this paper compared to [13] is the use of criteria based on stability as introduced in [2].

3.1 Fastly Cross-validated Kernel Machines

One way to decrease the degrees of freedom for c in the additive regularization LS-SVM used for fusion in a cross-validation context is to confine the regularization trade-off to be constant over the different folds.

As such, one gets a multi-criterion optimization problem

$$(w^{(l)}, e_k^{(l)}, b) = \underset{w^{(l)}, e_k^{(l)}, b}{\arg\min} \begin{bmatrix} \frac{1}{2} {w^{(1)}}^T w^{(1)} + \frac{1}{2} \sum_{k \in \mathcal{T}^{(1)}} (e_k^{(1)} - c_k)^2 \\ \dots \\ \frac{1}{2} {w^{(L)}}^T w^{(L)} + \frac{1}{2} \sum_{k \in \mathcal{T}^{(L)}} (e_k^{(L)} - c_k)^2 \end{bmatrix}$$
$$\text{s.t.} \quad \begin{cases} {w^{(1)}}^T \varphi(x_k) + b + e_k^{(1)} = y_k & \forall k \in \mathcal{T}^{(1)} \\ \dots \\ {w^{(L)}}^T \varphi(x_k) + b + e_k^{(L)} = y_k & \forall k \in \mathcal{T}^{(L)} \end{cases} \quad \text{holds.} \tag{16}$$

Let the solution to the final cross-validated learning machine take the following form

$$\hat{f}_{(cv)}(x) = \bar{w}^T \varphi(x) + b \quad \text{with} \quad \bar{w} = \frac{1}{L} \sum_{l=1}^{L} w^{(l)}. \tag{17}$$

Although the criteria of (16) can be solved individually but with coupled regularization constants [13], one can relax the problem by trying to find one Pareto-optimal solution [3]. The scalarization technique with weights $1_N = (1, \dots, 1)^T \in \mathbb{R}^L$ in the objective function is used leading to a much compacter problem than the original formulation [13]. The cost criterion becomes as such

$$J_P^{(cv)}\left(w^{(l)}, e_k^{(l)}\right) = \frac{1}{2(L-1)} \sum_{l=1}^{L} {w^{(l)}}^T w^{(l)} + \frac{1}{2} \sum_{k=1}^{N} (e_k - \tilde{c}_k)^2 \tag{18}$$

where $e_k = \frac{1}{L-1} \sum_{l|k \in \mathcal{T}^{(l)}} e_k^{(l)}$ and let $\tilde{c}$ be such that $\frac{1}{L-1} \sum_{l|k \in \mathcal{T}^{(l)}} \left(e_k^{(l)} - c_k\right)^2$ equals $(e_k - \tilde{c}_k)^2$ for all $k = 1, \dots, N$. Eliminating the residuals $e_k^{(l)}$, the the following constrained optimization approach to the cross-validation based AReg LS-SVM is obtained

$$\begin{aligned} \min_{w^{(l)}, b, e_k} \mathcal{J}^{(cv)} &= \frac{1}{2} \sum_{l=1}^{L} \frac{{w^{(l)}}^T w^{(l)}}{(L-1)} + \frac{1}{2} \sum_{k=1}^{N} (e_k - \tilde{c}_k)^2 \\ \text{s.t.} \quad & \frac{1}{L-1} \sum_{l|k \in \mathcal{T}_l} {w^{(l)}}^T \varphi(x_k) + b + e_k = y_k \quad \forall k = 1, \dots N. \end{aligned} \tag{19}$$

The Lagrangian of this constrained optimization problem becomes

$$\begin{aligned} \mathcal{L}^{(cv)}(w^{(l)}, b, e_k; \alpha_k) \quad = \quad & \frac{1}{2} \sum_{k=1}^{N} (e_k - \tilde{c}_k)^2 + \frac{1}{2} \sum_{l=1}^{L} \frac{{w^{(l)}}^T w^{(l)}}{L-1} \\ & - \sum_{k=1}^{N} \alpha_k \left(\frac{1}{L-1} \sum_{l|i \in \mathcal{T}_l} {w^{(l)}}^T \varphi(x_k) + b + e_k - y_k \right). \end{aligned} \tag{20}$$

The conditions for optimality w.r.t. $w^{(l)}, b, e_k, \alpha_k, \ \forall i, l$ for the training become:

$$\begin{cases} \partial \mathcal{L}^{(cv)} / \partial e_k = 0 & \rightarrow \quad e_k = \tilde{c}_k + \alpha_k & (a) \\ \partial \mathcal{L}^{(cv)} / \partial w^{(l)} = 0 & \rightarrow \quad w^{(l)} = \sum_{i \in \mathcal{T}_l} \alpha_k \varphi(x_k) & (b) \\ \partial \mathcal{L}^{(cv)} / \partial b = 0 & \rightarrow \quad \sum_{k=1}^{N} \alpha_k = 0 & (c) \\ \partial \mathcal{L}^{(cv)} / \partial \alpha_k = 0 & \rightarrow \quad \sum_{l|i \in \mathcal{T}_l} {w^{(l)}}^T \varphi(x_k) + b + e_k = y_k. & (d) \end{cases} \tag{21}$$

From (21.b) one can recover training equations in the dual space:

$$\sum_{l|i \in \mathcal{T}_l} w^{(l)} = \sum_{l|i \in \mathcal{T}_l} \sum_{i \in \mathcal{T}_l} \alpha_k \varphi(x_k) = (L-1) \sum_{k=1}^{N} \alpha_k \varphi(x_k) + \sum_{j \in \mathcal{V}_l} \alpha_j \varphi(x_j). \tag{22}$$

After elimination of the variables $w^{(l)}$ and $\tilde{c}$ this can be summarized in matrix notation as

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + \frac{1}{L-1}\Omega^{(cv)} \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline e \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y \end{array}\right] \tag{23}$$

with

$$\Omega^{(cv)} = \begin{bmatrix} \Omega^{\mathcal{V}_1} & & & \\ & \Omega^{\mathcal{V}_2} & & \\ & & \ddots & \\ & & & \Omega^{\mathcal{V}_L} \end{bmatrix} \tag{24}$$

and $\Omega^{\mathcal{V}_l} \in \mathbb{R}^{n^{(l)} \times n^{(l)}}$ is the kernel matrix between elements of the validation set of the lth fold $\Omega^{\mathcal{V}_l}_{i,j} = K(x_k, x_j), \forall i, j \in \mathcal{V}_l$. From (21.b) one can recover an expression for the individual models of the different folds such that the l-th model can be evaluated in point $x_j^{(v)}$ for $j \in \mathcal{V}_l$ as

$$y_j^{(v)} = (\hat{w}^{(l)})^T \varphi(x_j^v) + \hat{b} + e_j^{(v)} = \sum_{k \in \mathcal{I}_l} \hat{\alpha}_k K(x_k, x_j^v) + \hat{b} + e_j^{(v)} \tag{25}$$

with residual $\hat{f}^{(l)}(x_j^{(v)}) - y_j^{(v)}$ denoted as $e_j^{(v)}$ and $\hat{\alpha}$ and $\hat{b}$ solve (23). The global model (17) can also be recovered from (21.b) and can be evaluated in a new point x^* by

$$f^{(cv)}(x^*) = \bar{w}^T \varphi(x^*) = \sum_{k=1}^{N} \alpha_k \varphi(x_k) + b. \tag{26}$$

Now one has all elements to estimate a global model, while one still can assess the individual folds of the model. The fusion of the training equations (23) and the validation set of equations (25) results in the following constrained optimization problem

$$\textbf{Fusion}: \ (\hat{c}, \hat{\alpha}, \hat{b}) = \underset{c,\alpha,b}{\arg\min} \sum_{k=1}^{N} e_k^2 + \sum_{k=1}^{N} \left(e_k - e_k^{(v)}\right)^2 \quad \text{s.t.} \quad \text{(23) and (25) holds.} \tag{27}$$

This is equivalent to solving the following set of equations in unknowns α, b in least squares sense of e_k and $e_k - e_k^{(v)}$ and e^v:

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \frac{L-1}{L}\Omega + \frac{1}{L}\Omega^{(cv)} \\ \hline 0_N & \frac{L+1}{L}\Omega^{(cv)} - \frac{1}{L}\Omega \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline e \\ \hline e - e^v \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \\ \hline 0_N \end{array}\right]. \tag{28}$$

Let us now consider alternatives to (27) for creating stable kernel machines.

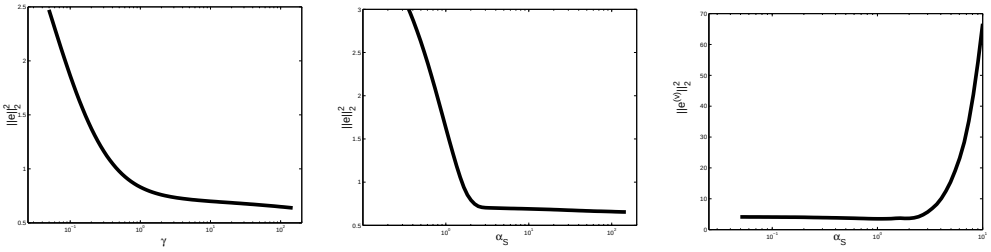
3.2 α-Stable Kernel Machines

While most stability criteria take a form based on the difference in loss between the training and leave-one-out error, a common relaxed version called α-stability can be taken

$$\left|e_k - e_k^{(v)}\right| \leq \alpha \quad \forall k = 1, \ldots, N. \tag{29}$$

This is considered as a measure for measuring the performance of learning machines and used to derive bounds on the generalization abilities. Here, we use it as a special form of regularization. Imposing $\alpha_{\mathcal{S}}$-stability on additively regularized (AReg) LS-SVMs boils down to a quadratic programming problem:

$$\mathcal{J}_{\alpha_{\mathcal{S}}}(e, e^{(v)}) = \|e\|_2^2 \quad \text{s.t.} \quad \left\{ \begin{array}{l} -\alpha_{\mathcal{S}} \leq e_k - e_k^{(v)} \leq \alpha_{\mathcal{S}} \quad \forall k = 1, \ldots, N \\ \text{(28) holds,} \end{array} \right. \tag{30}$$



(a) (b) (c)

Figure 3: *The toy problem as described in Section 4 was used to generate the following figures:* **(a)** *Classical L-curve of the regularization parameter γ in (4) with respect to the training error;* **(b)** *The L_α curve visualizing the trade-off between fitting error $\|e\|_2^2$ of (32) and the α upper bound of the stability measure;* **(c)** *The curve visualizing a typical relationship between validation errors of the leave-one-out formulation of (32) and the α upper bound of the stability measure.*

for a given value $\alpha_{\mathcal{S}}$. One can visualize the trade-off between stability and loss in a graph by exploring the solutions for a range of values of $\alpha_{\mathcal{S}}$. We shall refer to this graph as the L_α-curve, analogously to the L-curve [12, 9] displaying the trade-off between bias and variance (see Figure 3). According to the paradigm of fusion of training and validation (in this case stability maximization), the best performing model with maximal stability can be found by minimizing the following cost function

$$\mathcal{J}_\xi(e, e^{(v)}, \alpha_{\mathcal{S}}) = \|e\|_2^2 + \xi\alpha_{\mathcal{S}}, \quad \text{s.t.} \quad \begin{cases} -\alpha_{\mathcal{S}} \leq e_k - e_k^{(v)} \leq \alpha_{\mathcal{S}} \quad \forall k = 1, \ldots, N \\ (28) \text{ holds,} \end{cases} \tag{31}$$

which can be optimized using quadratic programming:

$$\min_{\alpha,b,\alpha_{\mathcal{S}}} \|\Omega\alpha + 1_N b - Y\|_2^2 + \xi\alpha_{\mathcal{S}} \text{ s.t. } -\alpha_{\mathcal{S}} 1_N \leq \left(\frac{L+1}{L}\Omega^{(cv)} - \frac{1}{L}\Omega\right)\alpha \leq \alpha_{\mathcal{S}} 1_N,\ 1_N^T\alpha = 0 \tag{32}$$

3.3 β-Stable Kernel Machines

Consider the β stability criterion as often used

$$\left|\ell(e_k) - \ell\left(e_k^{(v)}\right)\right| \leq \beta \quad \forall k = 1, \ldots, N. \tag{33}$$

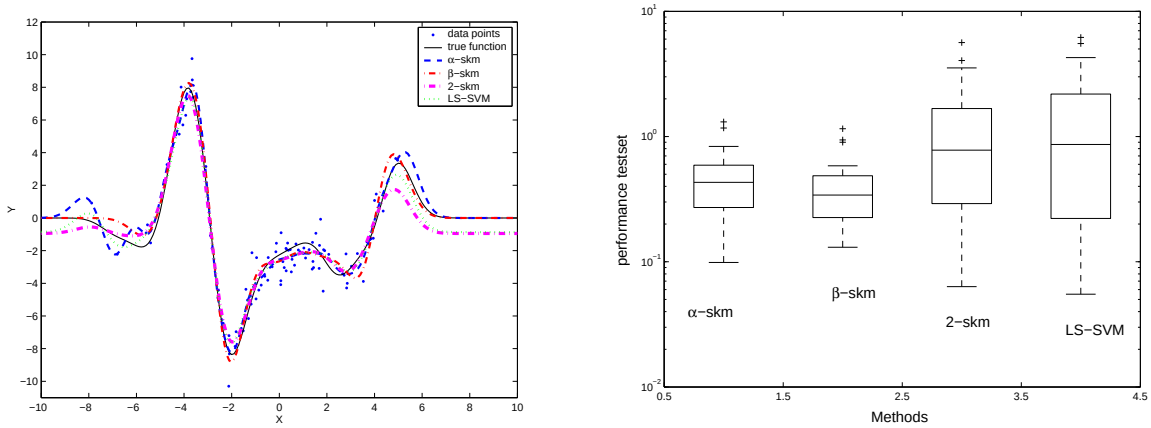
where in this case $\ell(e) = e^2$ is the squared loss function. Using this criterion to impose stability on a least squares SVM results in the following optimization problem.

$$\mathcal{J}_{\beta_{\mathcal{S}}}(e, e^{(v)}) = \|e\|_2^2 \quad \text{s.t.} \quad \begin{cases} -\beta_{\mathcal{S}} \leq e_k^2 - \left(e_k^{(v)}\right)^2 \leq \beta_{\mathcal{S}} \quad \forall k = 1, \ldots, N \\ (28) \text{ holds.} \end{cases} \tag{34}$$

According to the paradigm of fusion of training and validation (in this case stability maximization), the best performing model with maximal stability can be found by minimizing the following cost-function

$$\mathcal{J}_\xi(e, e^{(v)}, \beta_{\mathcal{S}}) = \|e\|_2^2 + \xi\beta_{\mathcal{S}}, \quad \text{s.t.} \quad \begin{cases} -\beta_{\mathcal{S}} \leq e_k^2 - \left(e_k^{(v)}\right)^2 \leq \beta_{\mathcal{S}} \quad \forall k = 1, \ldots, N \\ (28) \text{ holds.} \end{cases} \tag{35}$$

Using the variables β_k^+ and β_k^- satisfying $\beta_k^+ \geq |e_k + e_k^{(v)}|$ and $\beta_k^- \geq |e_k - e_k^{(v)}|$ for all $k = 1, \ldots, N$, this can be rewritten as a convex optimization problem which can be solved as a second order cone program-



(a) (b)

Figure 4: *Results from numerical experiments with the data generating mechanism as described in Section 4.* **(a)** *Result of the α-stable (32), α-stable (36), 2-norm (27) and standard LS-SVM on a particular realization of the dataset.* **(b)** *Boxplot of the obtained accuracy obtained on a testset on a Monte-carlo study of the different methods for randomly generated functions according to equation (37).*

ming problem.

$$\min_{\alpha,b,\beta^+,\beta^-} \|\Omega\alpha+1_N b-Y\|_2^2+\xi\beta_{\mathcal{S}} \text{ s.t. } \begin{cases} -\beta^- < \left(\frac{L+1}{L}\Omega^{(cv)} - \frac{1}{L}\Omega\right)\alpha \leq \beta^- \\ -\beta^+ < 1_N 2b + \left(\frac{2L-1}{L}\Omega - \frac{L-1}{L}\Omega^{(cv)}\right)\alpha - 2Y \leq \beta^+ \\ \beta^-\beta^+ \leq \beta_{\mathcal{S}}. \end{cases} \tag{36}$$

4 Illustrative Example

The experiments focus on the choice of the regularization scheme in kernel based models. For the design of a Monte-Carlo experiment, the choice of the kernel and kernel-parameter should not be of critical importance. To randomize the design of the underlying functions in the experiment with known kernel-parameter, the following class of functions is considered

$$f(\cdot) = \sum_{k=1}^{N} \bar{\alpha}_k K(x_k, \cdot) \tag{37}$$

where the input points x_k are equidistantly taken between 0 and 5 for all $k = 1, \ldots, N$ with $N = 75$ and $\bar{\alpha}_k$ is an i.i.d. uniformly randomly generated term. The kernel is fixed as $K(x_k, x_j) = \exp(-\|x_k - x_j\|_2^2)$ for all $i, j = 1, \ldots, N$. Output data points points were generated as $y_k = f(x_k) + e_k$ for $k = 1, \ldots, N$ where e_k are N i.i.d. samples of a Gaussian distribution.

Given this method to generate datasets with a prefixed kernel, a Monte Carlo study was conducted to relate the designed algorithms in a practical way as reported in Figure 4.

5 Conclusions

This paper extends results from the additive regularization framework and the fusion of training and cross-validation as introduced in [13]. After briefly reviewing those ideas, we proceed by designing a regularization scheme based on different stability measures. While from a conceptual point of view lthe oss function

and the stability measures are to be considered at a different levels, we show that from an optimization point of view they can be fused into one convex optimization problem. Furthermore, we also propose a new notion of L_α and L_β curves.

Acknowledgments. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. It is supported by grants from several funding agencies and sources: Research Council KU Leuven: Concerted Research Action GOA-Mefisto 666 (Mathematical Engineering), IDO (IOTA Oncology, Genetic networks), several PhD/postdoc & fellow grants; Flemish Government: Fund for Scientific Research Flanders (several PhD/postdoc grants, projects G.0407.02 (support vector machines), G.0256.97 (subspace), G.0115.01 (bio-i and microarrays), G.0240.99 (multilinear algebra), G.0197.02 (power islands), research communities ICCoS, ANMMM), AWI (Bil. Int. Collaboration Hungary/ Poland), IWT (Soft4s (softsensors), STWW-Genprom (gene promotor prediction), GBOU-McKnow (Knowledge management algorithms), Eureka-Impact (MPC-control), Eureka-FLiTE (flutter modeling), several PhD grants); Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) and IUAP V-10-29 (2002-2006) (2002-2006): Dynamical Systems and Control: Computation, Identification & Modelling), Program Sustainable Development PODO-II (CP/40: Sustainibility effects of Traffic Management Systems); Direct contract research: Verhaert, Electrabel, Elia, Data4s, IPCOS. JS is an associate professor and BDM is a full professor at K.U.Leuven Belgium, respectively.

References

[1] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] D. Calvetti, G. Golub, and L. Reichel. Estimation of the l-curve via lanczos bidiagonalization. *BIT. Nordisk Tidskrift for Informationsbehandling (BIT)*, 39(2):603–619, 1999.

[5] N. A. C. Cressie. *Statistics for spatial data*. Wiley, 1993.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[7] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[8] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, 1989.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, Heidelberg, 2001.

[11] D.J.C. MacKay. Introduction to gaussian processes. In C.M. Bishop, editor, *Neural networks and machine learning*, volume 168 of *Series F:Computer and Systems Sciences*, pages 133–165. Springer NATO-ASI, 1998.

[12] A. Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Review*, 40(3):636–666, 1998.

[13] K. Pelckmans, J.A.K. Suykens, and B. De Moor. Additive regularization: Fusion of training and validation levels in kernel methods. *(Submitted for Publication) Internal Report 03-184, ESAT-SISTA, K.U.Leuven (Leuven, Belgium)*, 2003.

[14] T. Poggio and F. Girosi. Networks for approximation and learning. In *Proceedings of the IEEE*, volume 78, pages 1481–1497, 1990.

[15] T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419 – 422, 2004.

[16] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proc. of the 15th Int. Conf. on Machine learning(ICML'98)*, pages 515–521. Morgan Kaufmann, 1998.

[17] B. Schoelkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[18] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistics Society Series*, B(36):111–147, 1974.

[19] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[20] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[21] A.N. Tikhonov and V.Y. Arsenin. *Solution of Ill-Posed Problems*. Winston, Washington DC, 1977.

[22] V.N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[23] G. Wahba. *Spline models for observational data*. SIAM, 1990.